Lección 25: Colisiones

Lección en línea ver en Code Studio

Propósito

Los estudiantes programan sus sprites para interactuar de nuevas maneras. Después de una breve revisión de cómo usaron el bloque isTouching, los estudiantes intercambian ideas sobre otras formas en que dos sprites podrían interactuar. A continuación, utilizan isTouching para hacer que un sprite empuje al otro a través de la pantalla antes de practicar con los cuatro bloques de colisión (collide, displace, bounce, y bounceOff).

Esta Lección introduce colisiones, otra abstracción útil que permitirá a los estudiantes manipular sus sprites de maneras completamente nuevas. Si bien los estudiantes teóricamente pueden haber escrito sus propios comandos de desplazamiento, colisión o rebote, la capacidad de ignorar los detalles de este código les permite centrar su atención en la estructura de alto nivel de los juegos que desean construir.

Esta Lección también pretende que los estudiantes practiquen utilizando los nuevos comandos que han aprendido. En realidad, es la última vez que aprenderán un nuevo comportamiento de sprite, y después de esta Lección, los estudiantes pasarán a centrarse más en cómo organizan su código cada vez más complejo.

Secuencia para el aprendizaje

- Conocimiento inicial (5 min)
- Ampliación del conocimiento (40 min)
- Transferencia del conocimiento (10 min)

Objetivos

Los estudiantes serán capaces de:

- Utilizar los bloques “displace”, “collide”, “rebunce”, y “bounceOff” para producir interacciones de sprites
- Describir cómo se pueden construir las abstracciones para desarrollar aún más abstracciones.

Vocabulario

Abstracción: una representación simplificada de algo más complejo. Las abstracciones le permiten ocultar detalles para ayudarlo a manejar la complejidad, enfocarse en conceptos relevantes y razonar sobre problemas en un nivel superior.

Código

- sprite.bounce(target)
- sprite.bounceOff(target)
- sprite.collide(target)
- sprite.displace(target)
- setCollider(type, xOffset, yOffset, width/radius, height, rotationOffset)
- sprite.bounciness

Estrategia de aprendizaje

Conocimiento inicial (5 min)

Pantalla:

Si tiene la capacidad, proyecte la animación en el primer nivel en Code Studio para esta Lección. De lo contrario, pida a las parejas de estudiantes que lo vean juntos. Muestre dos sprites, uno moviéndose a través de la pantalla hacia el otro y finalmente empujando uno cuando colisionan.

Resumen:

La clase programa sus sprites para interactuar de nuevas maneras. Después de una breve revisión de cómo usaron el bloque isTouching, la clase intercambia ideas sobre otras formas en que dos sprites podrían interactuar. Luego usan isTouching para hacer que un sprite presione otro en la pantalla, antes de practicar con los cuatro bloques de colisión (colisionar, desplazar, rebotar y rebotar).

Indicación:

Usando los bloques que ya sabemos cómo usar, ¿cómo podríamos crear la interacción de sprites que podemos ver en este programa?

Meta: El objetivo de esta discusión es que los estudiantes piensen en formasde resolver el problema de tener un elemento sprite empujando otro enla pantalla. No es necesario que los estudiantes lleguen a un consenso, ya que cada uno de ellos tendrá la oportunidad de probar una solución en el siguiente nivel en Code Studio. Los estudiantes deben entender quees posible usar bloques para producir el movimiento deseado solo con los bloques que ya han aprendido.

Tenemos muchas ideas geniales sobre cómo podríamos hacer que un sprite empuje otro en la pantalla. Ahora que ya se han preparado, puedes probar sus ideas en Code Studio. Una gran parte del problema es averiguar cuándo se tocan los dos sprites, pero como ya hemos descubierto cómo hacerlo y ahora podemos usar el bloque isTouching, ya no necesitamos pensarlo. Podemos enfocarnos en la nueva parte del problema.

Ampliación del conocimiento

Niveles de Code Studio

Niveles 2-4:

En estos niveles, se muestra a los estudiantes una interacción de sprites. Luego implementan sus ideas para crear la interacción de los sprites que observaron. La primera vez pueden implementar las ideas del grupo. La segunda vez desafíalos a implementar ese comportamiento de forma independiente.

Interacciones de Sprite

Hasta ahora han sido capaces de crear interacciones de sprites simples usando el sprite. bloque isTouching(). Por ejemplo, han reestablecido una moneda a una ubicación diferente en la pantalla cuando un personaje la toca. Ahora es el momento de comenzar a hacer que los sprites tengan interacciones más complejas.

Hacer esto

- Ejecuten el programa y observen la interacción entre los dos sprites.
- Discutan con un compañero: usando sólo los comandos que ya conocen, ¿cómo podrían crear este tipo de interacción? Hay muchas maneras de hacerlo, pero aquí hay algunos bloques a considerar:
- sprite.isTouching
- sprite.velocityX
- sprite.velocityY
- sprite.x
- sprite.y

Prepárense para compartir ideas con los compañeros de clase.

Inducción:

Éste fue un problema desafiante, pero pudimos resolverlo.

¿Qué nos ayudó a resolver este problema?

Todas estas cosas son muy importantes y surgen mucho en Informática. Una cosa que fue particularmente útil fue el isTouching bloque, que ocultó el código complicado que nos dice si los dos sprites se están tocando. También hay un bloque (displace) que oculta el código que acabamos de escribir, y algunos otros bloques que ocultan el código para otros tipos de interacciones de sprites. Tendrás la oportunidad de probar estos bloques en los siguientes niveles y utilizarlos para mejorar tu juego de vuelo.

Niveles 5-10:

Estos niveles introducen cómo usar los 4 nuevos bloques de colisión que los estudiantes aprenderán en esta Lección.

Desplazar

sprite.displace hará que un sprite empuje al otro cuando se toquen.

Tipos de colisión

Hay cuatro tipos de colisiones que usamos en Game Lab. Estos bloques causarán un cierto tipo de interacción entre el sprite y su objetivo.

Desplazar

El bloque displace hace que el sprite empuje al objetivo siempre que se toquen entre sí.

El sprite se mueve normalmente.

Chocar

El bloque collide hace que el sprite se detenga cuando se encuentra con el objetivo. Si el objetivo se está moviendo, empujará al objeto con él. El objetivo sigue moviéndose normalmente.

Este es un buen momento para decir cuánto han progresado los estudiantes en sus habilidades desde el comienzo de la unidad. Este problema habría parecido casi imposible a principios de año. Algunas cosas que hicieron que elproblema fuera más fácil de resolver fueron:

- Preparación: Los estudiantes intercambiaron ideas y pensaron en soluciones antes de probar sucódgo
- Cooperación: Los estudiantes trabajaron en grupos para llegar auna solución
- Abstracción: los estudiantes pudieron usar los bloques **isTouching** y **velocityY** para ocultar parte de la complejidad de la solución

Rebotar

El bloque bounce hace que el sprite y el objetivo reboten cuando se tocan entre sí. Tanto el sprite como el objetivo cambian cómo se están moviendo. El bloque bounceOff hace que el sprite rebote en el objetivo. El objetivo sigue moviéndose normalmente.

Depurar

Hay una sprite.debug propiedad especial que puede usar para comprender mejor por qué los sprites interactúan de la manera en que lo hacen.

SetCollider

Los Sprites interactúan en función del tamaño y la forma de su colisionador, no de las imágenes que se les asignan. Solo se puede ver el colisionador cuando se activa el modo de depuración. Puede cambiar la forma del colisionador usando el sprite.setCollider() bloque, que le permite elegir entre un “rectángulo” o un “círculo”. Por defecto, todos los colisionadores son “rectangulares”.

Niveles 11-15:

Estos niveles guían a los estudiantes al agregar colisiones al juego que comenzaron en la Lección anterior, y eventualmente invitan a los estudiantes a agregar sus propias modificaciones al juego.

Bounciness

Hasta el momento, bounceOff ha hecho que los sprites se alejen de otros objetos tan rápido como rebotaban en ellos. En el mundo real, casi todo se ralentiza un poco cuando rebota en otra cosa. Puede usar el bounciness bloque para decirle a tu sprite cuánto desacelerar o acelerar cuando rebota en otra cosa.

Transferencia del conocimiento (10 min)

Compartir y publicar 3-2-1

De a los estudiantes tiempo para jugar los juegos de los demás. Pida que se centren no sólo en el nuevo comportamiento que agregaron sino también en el código que usaron para crearlo.

Revisar:

Pida a los estudiantes que escriban y reflexionen sobre las siguientes indicaciones.

- ¿Tres cosas que viste en el juego de otra persona que realmente te gustaron?
- ¿Cuáles son dos mejoras que harías en tu juego si tuvieras más tiempo?
- ¿Cuál es un bloque que te gustaría tener en Game Lab y cómo funcionaría?

Sugerencias para evaluación

Se sugiere el siguiente indicador para evaluar formativamente los aprendizajes:

- Utilizan bloques para programar en base un diseño predictivo
- Explican la forma en que diseñan su programa.